On the Iota-Delta Function: Mathematical Representation of Two-Dimensional Cellular Automata

Luan C. de S. M. Ozelim*
André L. B. Cavalcante†
Department of Civil and Environmental Engineering
University of Brasilia
Brasilia, DF, 70910-900, Brazil
**luanoz@gmail.com*
†abrasil@unb.br. Full address for mailing: Geotecnia - Departamento de Engenharia Civil e Ambiental/FT - UnB, 70910-900 - Brasília/DF/Brazil

Even though the patterns appearing in the evolution of two-dimensional cellular automata have been deeply studied, the evolution rules themselves have not received the same amount of attention. In the present paper, the evolution rules of totalistic and outer totalistic two-dimensional cellular automata for a set of neighborhood templates have been expressed in terms of the iota-delta function. Additionally, the idea of iota-delta function bases for the rule space of two-dimensional cellular automata evolution is introduced. By means of suitable bases, every two-dimensional cellular automaton can have its evolution rule described in terms of the iota-delta function. This approach enables investigating the evolution rules as mathematical functions and the evolution itself as a function composition process.

1. Introduction

By definition, a cellular automaton (CA) consists of a neighborhood template N, a state space S, and a state transition function f that updates the state space [1, 2].

One of the most studied topics in CA theory is the behavior of the state space. For example, the patterns generated by the evolution of a CA such as Conway's Game of Life [3] have been deeply studied and classified as gliders, beehives, loafs, and so on.

The evolution rules, on the other hand, tend to receive less attention in the process; that is, the state transition functions are taken into account in order to implement the CA, but no deep analysis has been performed on them.

By drawing a parallel between discrete and continuous dynamical systems, the state transition functions of the former correspond to the

Complex Systems, 22 © 2013 Complex Systems Publications, Inc.
https://doi.org/10.25088/ComplexSystems.22.4.405

partial differential equations of the latter. It is widely known that the partial differential equations that govern a given phenomenon carry important information about continuous dynamical systems. In fact, the presence of first- and higher-order derivatives is often related to advective and dispersive phenomena, respectively.

In order to represent the state transition functions, several approaches have been proposed, namely: lookup tables [1], Boolean algebra [1], algebraic equations [1], and genetic algorithms [4]. It is clear that in order to proceed to a rigorous mathematical treatment of the evolution rules, it is of interest that the state transition functions are in fact mathematical functions.

Using Boolean algebra is interesting for dealing with one-dimensional binary systems. As the number of dimensions grows, the Boolean expressions tend to become highly intricate, ultimately preventing a full mathematical analysis. Algebraic equations, on the other hand, are of great interest but can only be assigned to a few CAs. A recent paper [5] introduced a new special function, called the iota-delta function, that is able to straightforwardly present the state transition functions of CAs. It is shown in [5] that every elementary CA can have its evolution rules represented in terms of the iota-delta function.

In the present paper, the propositions from [5] are further developed in order to represent not only the state transition function of every elementary CA but also every two-dimensional CA. This way, it is possible to establish a solid mathematical framework for the study of evolution rules.

Section 2 discusses the iota-delta function.

2. The Iota-Delta Function

Let the iota-delta function be defined as follows [5]:

$$\iota\delta_n^m[x] = \text{mod}\big[\text{mod}\big[\ldots\text{mod}[\text{mod}[x, p_m], p_{m-1}], \ldots, p_j\big], n\big],$$
$$m \geq j; \quad m, n \in \mathbb{Z}_+; \quad x \in \mathbb{C}; \quad j = \pi[n] + 1; \tag{1}$$

in which $\text{mod}[o, p]$ denotes the modulus operator, which gives the rest of the division of o by p if o is greater than p or o itself; otherwise, m and n are parameters of the iota-delta function, p_m is the m^{th} prime number, and $\pi[n]$ stands for the prime counting function that gives the number of primes less than or equal to n. Note that it is considered that $p_1 = 2$. The value of n determines how many states the automata generated have. Thus, for binary automata $n = 2$, for ternary ones $n = 3$, for quaternary ones $n = 4$, and so on. Also, the

https://doi.org/10.25088/ComplexSystems.22.4.405

iota-delta function is taken to be non-negative and $\max[\iota\delta_n^m[x]] \to n$ when $x \in \mathbb{R}$ [5]. The iota-delta function is periodic with a period length of p_m.

2.1 The Iota-Delta Function and Elementary Cellular Automata

As stated in Section 1, it has been shown in [5] that every elementary CA can be represented by the following evolution rule:

$$C_k^{i+1} = \iota\delta_2^m \left[\alpha_1\, C_{k-1}^i + \alpha_2\, C_k^i + \alpha_3\, C_{k+1}^i + \alpha_4\right], \tag{2}$$

in which the coefficients are $\alpha_j = \{r \mid r \leq p_m - 1;\; r \in \mathbb{Z}_+\}$ and $j = 1, 2, 3, 4$. In the case of elementary CAs, the minimum value of m that enables expressing every rule is $m = 5$; that is, $\iota\delta_2^5[x] = \text{mod}[\text{mod}[\text{mod}[\text{mod}[\text{mod}[x, 11], 7], 5], 3], 2]$. Also, the inferior and superior indexes k and i, respectively, are related to the position of the cell in the bidimensional (space and time) state space of the CA.

Each elementary CA has a set of tuples $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ that characterizes it. A complete list of tuples can be found in [5]. For example, the well-known rule 30 can be described by this simple updating rule [5]:

$$C_k^{i+1} = \iota\delta_2^5 \left[C_{k-1}^i + 4\, C_k^i + 4\, C_{k+1}^i\right]. \tag{3}$$

Sections 3 through 5 develop the framework for obtaining the evolution rules of two-dimensional CAs.

3. The Iota-Delta Function and Totalistic Two-Dimensional Cellular Automata

The philosophy behind using the iota-delta function for representing any evolution rule is recognizing which variables are important for determining the value of a given cell. For example, in the case of elementary CAs, the value of a given cell depends on the value of its immediate left and right neighbors and the value of the cell itself in the current step.

This elementary case shows that three variables determine the value of the cell of interest in the next step. This way, the iota-delta function representation given in [5] and reproduced in equation (2) fits the need. It is worth noticing that besides the coefficients multiplying each variable, a fourth coefficient is added to enable representing odd rules (rule parity is determined by the output of the transformation when the three cells are 0).

Given these first thoughts about the iota-delta representation of evolution rules, it is necessary to investigate on which variables the

https://doi.org/10.25088/ComplexSystems.22.4.405

values of the cells of interest depend. In the case of CAs, the number of variables is deeply related to the neighborhood template considered. Also, the type of rule is of interest (totalistic, outer totalistic, ordinary, etc.). In the present paper, at first, special attention is given to totalistic and outer totalistic CAs. Following that, ordinary two-dimensional CAs are taken into account.

Totalistic CAs are, by definition, CAs in which the value of a given cell in the next time step is uniquely determined by the sum of the cells in its neighborhood template, including in the sum the cell itself [1]. This shows that, despite the number of cells in the neighborhood, the value of a given cell in the next time step is dependent on a single variable.

In Sections 3.1 and 3.2, every evolution rule of totalistic two-dimensional CAs for a set of neighborhood templates is shown to be expressible in terms of the iota-delta function.

3.1 Two-Dimensional Totalistic Cellular Automata with the Neumann Neighborhood

Consider the neighborhood template known as the Neumann neighborhood, graphically shown in Figure 1.

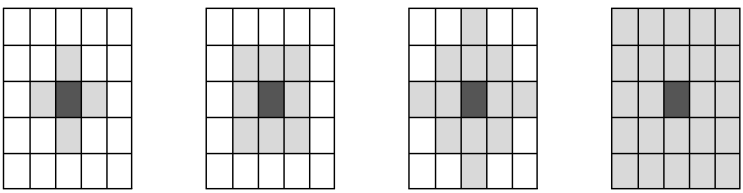

Figure 1. Schemes of Neumann, Moore, extended Neumann, and extended Moore neighborhood templates (from left to right).

As discussed, an appropriate iota-delta evolution rule for a two-dimensional totalistic CA is of the type:

$$C_{k,j}^{i+1} = \iota\delta_2^m [\alpha_1 \sigma + \alpha_2], \tag{4}$$

in which j is the additional spatial dimension and σ is the sum of the neighborhood cells and the cell of interest itself. In the case of a Neumann neighborhood template, the sum goes from 0 to 5. This makes a total of 2^6 totalistic rules in a Neumann neighborhood template. Each of these rules can be numbered following the scheme proposed in [1], in which the outputs of the transformation correspond to a coefficient in the binary decomposition of the rule number (abbreviated as RN in the tables). Table 1 shows the value of m, α_1, and α_2 as in equation (4) for every two-dimensional totalistic CA with the Neumann neighborhood.

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
0	{1, 0, 0}	16	{5, 7, 0}	32	{5, 9, 0}	48	{4, 5, 2}
1	{5, 2, 1}	17	{5, 4, 1}	33	{4, 2, 1}	49	{3, 2, 1}
2	{5, 8, 0}	18	{2, 1, 0}	34	{5, 10, 2}	50	{4, 4, 2}
3	{4, 2, 6}	19	{4, 3, 1}	35	{3, 3, 1}	51	{6, 6, 8}
4	{5, 6, 3}	20	{3, 1, 2}	36	{2, 2, 0}	52	{5, 3, 0}
5	{5, 1, 6}	21	{1, 1, 1}	37	{3, 4, 1}	53	{7, 3, 6}
6	{3, 2, 2}	22	{5, 8, 7}	38	{4, 4, 0}	54	{6, 11, 3}
7	{4, 5, 1}	23	{5, 2, 4}	39	{8, 11, 1}	55	{6, 2, 4}
8	{5, 5, 0}	24	{3, 3, 2}	40	{5, 10, 0}	56	{4, 5, 0}
9	{2, 1, 1}	25	{4, 3, 6}	41	{3, 1, 1}	57	{8, 11, 6}
10	{3, 4, 2}	26	{5, 3, 3}	42	{1, 1, 0}	58	{5, 9, 3}
11	{5, 8, 4}	27	{6, 2, 6}	43	{7, 14, 4}	59	{6, 11, 1}
12	{3, 2, 0}	28	{4, 2, 0}	44	{5, 8, 10}	60	{9, 17, 13}
13	{5, 3, 6}	29	{5, 9, 1}	45	{7, 15, 1}	61	{10, 3, 18}
14	{4, 2, 2}	30	{9, 6, 0}	46	{5, 2, 2}	62	{10, 6, 0}
15	{9, 17, 1}	31	{10, 23, 1}	47	{10, 26, 4}	63	{1, 0, 1}

Table 1. Iota-delta tuples $\{m, \alpha_1, \alpha_2\}$ for totalistic CAs with the Neumann neighborhood.

3.2 Two-Dimensional Totalistic Cellular Automata with the Moore Neighborhood

Consider now the neighborhood template known as the Moore neighborhood, graphically shown in Figure 1.

As in the case of a Neumann neighborhood, equation (4) is also applicable. In the case of a Moore neighborhood, on the other hand, σ goes from 0 to 9, making a total of 2^{10} totalistic rules. The same numbering scheme described in Section 3.1 can be applied. Table 2 gives the values of m, α_1, and α_2 as in equation (4) for every two-dimensional totalistic CA with the Moore neighborhood. It is worth noting that the values of m, α_1, and α_2 in the tables are the first ones that allow representing the evolution of the indicated CA rules in terms of equation (4). In this sense, the values in the tables are the smallest values that allow such a representation.

https://doi.org/10.25088/ComplexSystems.22.4.405

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
0	{1, 0, 0}	36	{7, 9, 7}	72	{7, 8, 11}	108	{6, 8, 3}
1	{7, 2, 1}	37	{5, 8, 1}	73	{5, 6, 1}	109	{8, 7, 4}
2	{7, 2, 16}	38	{6, 6, 0}	74	{5, 10, 7}	110	{6, 2, 2}
3	{7, 15, 6}	39	{8, 11, 1}	75	{5, 8, 4}	111	{12, 8, 27}
4	{9, 6, 15}	40	{6, 9, 0}	76	{6, 6, 7}	112	{4, 5, 2}
5	{7, 2, 14}	41	{6, 10, 8}	77	{6, 5, 4}	113	{5, 7, 6}
6	{6, 2, 10}	42	{8, 16, 2}	78	{8, 11, 9}	114	{11, 5, 7}
7	{6, 11, 8}	43	{10, 25, 18}	79	{15, 41, 24}	115	{10, 17, 20}
8	{8, 6, 5}	44	{7, 13, 9}	80	{6, 4, 3}	116	{5, 9, 5}
9	{10, 27, 4}	45	{7, 2, 8}	81	{6, 9, 4}	117	{12, 3, 12}
10	{7, 14, 0}	46	{5, 2, 2}	82	{6, 12, 5}	118	{6, 11, 3}
11	{5, 2, 6}	47	{13, 6, 12}	83	{10, 18, 6}	119	{14, 37, 1}
12	{9, 11, 2}	48	{8, 5, 0}	84	{5, 1, 2}	120	{9, 6, 11}
13	{6, 2, 8}	49	{9, 8, 20}	85	{10, 26, 24}	121	{11, 19, 6}
14	{5, 9, 10}	50	{9, 11, 3}	86	{10, 25, 22}	122	{10, 3, 15}
15	{9, 17, 1}	51	{6, 7, 12}	87	{13, 13, 1}	123	{12, 29, 1}
16	{9, 6, 3}	52	{5, 3, 0}	88	{7, 13, 13}	124	{10, 23, 13}
17	{7, 7, 12}	53	{7, 3, 6}	89	{7, 12, 4}	125	{14, 37, 42}
18	{7, 5, 9}	54	{6, 8, 11}	90	{8, 5, 3}	126	{17, 3, 45}
19	{6, 5, 1}	55	{6, 2, 4}	91	{7, 2, 6}	127	{23, 71, 54}
20	{7, 14, 3}	56	{4, 5, 0}	92	{5, 2, 0}	128	{9, 17, 0}
21	{8, 16, 18}	57	{10, 17, 8}	93	{11, 25, 1}	129	{8, 13, 8}
22	{7, 13, 5}	58	{5, 9, 3}	94	{14, 31, 13}	130	{7, 12, 13}
23	{5, 2, 4}	59	{6, 11, 1}	95	{13, 6, 6}	131	{6, 8, 4}
24	{10, 17, 2}	60	{9, 17, 13}	96	{10, 23, 5}	132	{12, 21, 13}
25	{9, 11, 14}	61	{10, 3, 18}	97	{8, 14, 12}	133	{9, 5, 8}
26	{7, 3, 9}	62	{10, 6, 0}	98	{7, 11, 10}	134	{5, 3, 9}
27	{6, 2, 6}	63	{17, 3, 48}	99	{3, 3, 1}	135	{5, 7, 1}
28	{7, 4, 13}	64	{8, 13, 2}	100	{5, 5, 9}	136	{7, 10, 10}
29	{5, 9, 1}	65	{7, 11, 6}	101	{9, 16, 20}	137	{7, 7, 8}
30	{9, 17, 7}	66	{9, 8, 16}	102	{6, 7, 5}	138	{5, 1, 5}
31	{10, 23, 1}	67	{5, 7, 8}	103	{8, 8, 4}	139	{8, 12, 6}
32	{9, 17, 11}	68	{7, 7, 15}	104	{7, 4, 11}	140	{5, 5, 2}
33	{9, 6, 20}	69	{13, 12, 14}	105	{5, 3, 8}	141	{11, 7, 6}
34	{7, 7, 5}	70	{7, 6, 2}	106	{7, 3, 3}	142	{5, 4, 0}
35	{5, 4, 8}	71	{5, 4, 4}	107	{14, 8, 27}	143	{12, 14, 27}

Table 2. (*continues*).

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
144	{7, 8, 3}	180	{8, 14, 5}	216	{6, 5, 10}	252	{20, 59, 7}
145	{5, 5, 6}	181	{11, 26, 24}	217	{6, 8, 8}	253	{17, 3, 42}
146	{2, 1, 0}	182	{8, 12, 15}	218	{8, 7, 16}	254	{23, 12, 53}
147	{5, 6, 6}	183	{7, 2, 4}	219	{8, 2, 6}	255	{23, 15, 65}
148	{7, 5, 11}	184	{5, 2, 9}	220	{6, 2, 0}	256	{7, 15, 0}
149	{5, 10, 8}	185	{13, 19, 14}	221	{10, 13, 4}	257	{9, 17, 6}
150	{5, 8, 7}	186	{11, 25, 7}	222	{12, 8, 19}	258	{11, 18, 0}
151	{15, 10, 4}	187	{11, 27, 24}	223	{15, 6, 6}	259	{5, 9, 6}
152	{5, 6, 10}	188	{13, 26, 31}	224	{7, 6, 10}	260	{7, 5, 2}
153	{4, 3, 6}	189	{17, 31, 57}	225	{4, 2, 1}	261	{7, 12, 1}
154	{10, 4, 2}	190	{13, 6, 0}	226	{5, 7, 10}	262	{6, 8, 9}
155	{6, 5, 12}	191	{17, 56, 4}	227	{7, 11, 14}	263	{7, 11, 1}
156	{8, 11, 17}	192	{9, 11, 0}	228	{8, 8, 2}	264	{9, 15, 19}
157	{10, 27, 12}	193	{10, 18, 14}	229	{11, 14, 27}	265	{9, 18, 12}
158	{20, 61, 34}	194	{6, 6, 11}	230	{8, 11, 3}	266	{9, 8, 0}
159	{15, 41, 30}	195	{12, 26, 35}	231	{11, 12, 8}	267	{10, 24, 6}
160	{7, 3, 10}	196	{5, 6, 3}	232	{5, 9, 7}	268	{6, 7, 0}
161	{6, 3, 4}	197	{11, 3, 14}	233	{15, 44, 6}	269	{5, 3, 6}
162	{6, 1, 7}	198	{3, 2, 2}	234	{11, 3, 9}	270	{5, 7, 5}
163	{6, 9, 8}	199	{7, 6, 6}	235	{15, 38, 4}	271	{11, 23, 20}
164	{7, 12, 5}	200	{6, 7, 9}	236	{6, 11, 5}	272	{7, 10, 0}
165	{3, 4, 1}	201	{4, 4, 1}	237	{7, 15, 1}	273	{9, 10, 18}
166	{7, 5, 13}	202	{10, 11, 16}	238	{14, 6, 2}	274	{5, 10, 5}
167	{14, 39, 12}	203	{9, 16, 4}	239	{16, 6, 12}	275	{7, 7, 1}
168	{5, 10, 0}	204	{6, 6, 9}	240	{9, 17, 2}	276	{6, 12, 3}
169	{4, 1, 1}	205	{14, 9, 24}	241	{12, 14, 8}	277	{5, 1, 4}
170	{14, 14, 0}	206	{8, 8, 15}	242	{16, 41, 39}	278	{8, 12, 13}
171	{9, 20, 4}	207	{14, 12, 6}	243	{11, 19, 18}	279	{13, 19, 30}
172	{7, 14, 10}	208	{7, 5, 5}	244	{13, 15, 19}	280	{7, 6, 7}
173	{11, 5, 20}	209	{9, 14, 14}	245	{10, 3, 12}	281	{5, 5, 8}
174	{13, 13, 29}	210	{5, 3, 5}	246	{12, 29, 9}	282	{10, 11, 26}
175	{15, 13, 35}	211	{11, 13, 14}	247	{17, 6, 18}	283	{11, 7, 30}
176	{5, 8, 5}	212	{7, 3, 0}	248	{10, 6, 17}	284	{5, 4, 7}
177	{10, 18, 27}	213	{9, 3, 6}	249	{18, 37, 12}	285	{10, 21, 1}
178	{7, 12, 9}	214	{14, 8, 19}	250	{14, 37, 5}	286	{12, 34, 7}
179	{14, 24, 8}	215	{15, 9, 35}	251	{15, 41, 1}	287	{16, 47, 30}

Table 2. (*continues*).

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
288	{7, 12, 3}	324	{5, 10, 3}	360	{8, 14, 10}	396	{3, 2, 0}
289	{5, 3, 4}	325	{6, 12, 1}	361	{8, 5, 12}	397	{14, 35, 8}
290	{5, 1, 7}	326	{10, 11, 19}	362	{11, 26, 29}	398	{7, 6, 0}
291	{5, 5, 1}	327	{6, 9, 12}	363	{17, 19, 8}	399	{12, 29, 14}
292	{2, 2, 0}	328	{5, 1, 9}	364	{8, 12, 3}	400	{6, 7, 2}
293	{9, 18, 14}	329	{9, 5, 20}	365	{9, 2, 8}	401	{7, 10, 6}
294	{5, 6, 0}	330	{3, 4, 2}	366	{7, 2, 2}	402	{5, 5, 10}
295	{17, 57, 8}	331	{6, 10, 4}	367	{19, 15, 12}	403	{4, 3, 1}
296	{6, 1, 9}	332	{10, 18, 28}	368	{5, 2, 7}	404	{7, 12, 7}
297	{3, 1, 1}	333	{7, 5, 8}	369	{10, 8, 24}	405	{17, 52, 32}
298	{4, 6, 2}	334	{11, 17, 15}	370	{11, 28, 11}	406	{9, 16, 11}
299	{7, 14, 4}	335	{16, 12, 8}	371	{13, 11, 24}	407	{13, 26, 27}
300	{5, 8, 10}	336	{8, 3, 13}	372	{11, 25, 13}	408	{6, 6, 3}
301	{8, 14, 14}	337	{5, 10, 1}	373	{21, 54, 30}	409	{6, 7, 4}
302	{15, 3, 29}	338	{4, 1, 0}	374	{11, 27, 28}	410	{14, 9, 15}
303	{19, 14, 57}	339	{11, 17, 18}	375	{20, 37, 48}	411	{16, 29, 51}
304	{9, 12, 10}	340	{14, 14, 29}	376	{10, 26, 13}	412	{8, 8, 7}
305	{5, 6, 4}	341	{1, 1, 1}	377	{14, 23, 18}	413	{13, 30, 30}
306	{4, 4, 2}	342	{9, 20, 7}	378	{17, 28, 10}	414	{14, 20, 7}
307	{6, 6, 8}	343	{10, 26, 1}	379	{20, 6, 24}	415	{20, 41, 57}
308	{7, 5, 15}	344	{7, 14, 13}	380	{14, 6, 37}	416	{7, 5, 0}
309	{10, 4, 27}	345	{10, 25, 1}	381	{13, 35, 1}	417	{8, 14, 1}
310	{6, 5, 7}	346	{11, 5, 15}	382	{17, 56, 7}	418	{8, 7, 7}
311	{10, 13, 1}	347	{12, 20, 18}	383	{26, 86, 87}	419	{12, 26, 1}
312	{11, 26, 21}	348	{11, 28, 5}	384	{6, 11, 2}	420	{5, 3, 2}
313	{8, 11, 6}	349	{20, 28, 1}	385	{7, 14, 8}	421	{6, 3, 6}
314	{11, 3, 15}	350	{10, 26, 7}	386	{6, 5, 3}	422	{9, 7, 17}
315	{10, 27, 14}	351	{20, 56, 57}	387	{8, 15, 8}	423	{11, 13, 1}
316	{16, 12, 37}	352	{7, 14, 2}	388	{5, 8, 3}	424	{10, 4, 15}
317	{20, 37, 32}	353	{5, 8, 8}	389	{6, 10, 12}	425	{7, 3, 14}
318	{15, 41, 36}	354	{10, 18, 9}	390	{12, 26, 9}	426	{9, 3, 3}
319	{22, 38, 30}	355	{12, 29, 35}	391	{7, 13, 12}	427	{16, 37, 1}
320	{7, 3, 7}	356	{10, 25, 9}	392	{7, 11, 5}	428	{14, 35, 5}
321	{10, 8, 18}	357	{7, 12, 14}	393	{5, 6, 8}	429	{17, 13, 12}
322	{9, 18, 2}	358	{14, 34, 10}	394	{10, 18, 2}	430	{15, 9, 26}
323	{6, 3, 1}	359	{14, 24, 27}	395	{14, 34, 4}	431	{22, 15, 42}

Table 2. (*continues*).

https://doi.org/10.25088/ComplexSystems.22.4.405

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
432	{6, 5, 5}	468	{13, 28, 23}	504	{17, 56, 13}	540	{4, 2, 0}
433	{11, 24, 24}	469	{11, 3, 6}	505	{22, 41, 18}	541	{12, 13, 12}
434	{6, 8, 0}	470	{15, 38, 13}	506	{17, 3, 39}	542	{9, 6, 0}
435	{13, 18, 24}	471	{21, 50, 27}	507	{25, 82, 35}	543	{16, 12, 6}
436	{8, 7, 9}	472	{6, 11, 7}	508	{23, 12, 41}	544	{7, 11, 0}
437	{14, 38, 24}	473	{10, 16, 18}	509	{35, 15, 35}	545	{5, 7, 4}
438	{18, 53, 9}	474	{7, 15, 3}	510	{23, 68, 19}	546	{9, 13, 16}
439	{8, 2, 4}	475	{8, 17, 1}	511	{30, 98, 72}	547	{9, 10, 8}
440	{6, 2, 11}	476	{14, 37, 13}	512	{7, 15, 2}	548	{5, 6, 7}
441	{10, 2, 27}	477	{18, 6, 57}	513	{9, 3, 4}	549	{5, 10, 6}
442	{11, 4, 23}	478	{22, 76, 7}	514	{9, 6, 21}	550	{7, 7, 11}
443	{10, 16, 8}	479	{16, 6, 6}	515	{6, 11, 6}	551	{11, 14, 4}
444	{12, 8, 11}	480	{9, 6, 22}	516	{8, 6, 11}	552	{6, 4, 7}
445	{23, 10, 57}	481	{9, 17, 8}	517	{5, 2, 8}	553	{6, 12, 4}
446	{15, 6, 0}	482	{12, 3, 17}	518	{7, 3, 15}	554	{5, 1, 3}
447	{23, 68, 57}	483	{12, 8, 24}	519	{5, 9, 8}	555	{11, 9, 18}
448	{5, 2, 3}	484	{20, 10, 15}	520	{7, 7, 2}	556	{9, 9, 2}
449	{7, 6, 4}	485	{16, 41, 51}	521	{6, 5, 6}	557	{8, 12, 1}
450	{5, 4, 2}	486	{14, 23, 15}	522	{10, 21, 3}	558	{10, 8, 19}
451	{4, 2, 6}	487	{11, 19, 30}	523	{7, 12, 6}	559	{14, 15, 12}
452	{5, 7, 3}	488	{14, 28, 3}	524	{10, 11, 2}	560	{9, 15, 0}
453	{6, 4, 6}	489	{13, 15, 4}	525	{6, 8, 1}	561	{3, 2, 1}
454	{7, 11, 3}	490	{10, 3, 9}	526	{7, 11, 7}	562	{5, 5, 3}
455	{8, 6, 8}	491	{24, 83, 1}	527	{10, 6, 12}	563	{7, 9, 12}
456	{8, 8, 13}	492	{12, 29, 17}	528	{9, 17, 5}	564	{10, 11, 15}
457	{17, 2, 51}	493	{19, 59, 54}	529	{5, 4, 1}	565	{14, 35, 24}
458	{11, 14, 13}	494	{22, 3, 59}	530	{5, 8, 9}	566	{11, 7, 23}
459	{11, 18, 12}	495	{17, 6, 12}	531	{6, 6, 6}	567	{12, 17, 1}
460	{8, 11, 11}	496	{10, 6, 11}	532	{6, 10, 5}	568	{5, 4, 3}
461	{13, 23, 27}	497	{13, 29, 8}	533	{10, 5, 4}	569	{12, 26, 8}
462	{16, 33, 5}	498	{15, 6, 29}	534	{8, 5, 13}	570	{10, 21, 9}
463	{11, 12, 27}	499	{20, 30, 30}	535	{14, 10, 4}	571	{11, 29, 14}
464	{5, 9, 9}	500	{13, 35, 13}	536	{8, 14, 2}	572	{12, 34, 10}
465	{10, 21, 4}	501	{23, 6, 65}	537	{6, 7, 6}	573	{12, 29, 30}
466	{15, 44, 9}	502	{15, 41, 7}	538	{5, 3, 3}	574	{13, 12, 23}
467	{20, 27, 48}	503	{16, 47, 1}	539	{12, 8, 6}	575	{21, 38, 24}

Table 2. (*continues*).

https://doi.org/10.25088/ComplexSystems.22.4.405

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
576	{10, 5, 11}	612	{4, 3, 0}	648	{13, 14, 11}	684	{9, 20, 10}
577	{6, 8, 12}	613	{11, 11, 8}	649	{5, 10, 4}	685	{12, 7, 6}
578	{9, 5, 13}	614	{6, 6, 2}	650	{6, 12, 2}	686	{11, 28, 2}
579	{5, 3, 1}	615	{10, 12, 6}	651	{14, 10, 8}	687	{10, 26, 4}
580	{7, 10, 3}	616	{7, 5, 10}	652	{11, 28, 10}	688	{7, 14, 16}
581	{5, 1, 6}	617	{11, 13, 27}	653	{10, 11, 8}	689	{14, 8, 38}
582	{5, 5, 7}	618	{10, 4, 23}	654	{6, 9, 3}	690	{10, 25, 5}
583	{12, 10, 4}	619	{17, 25, 35}	655	{12, 34, 4}	691	{14, 34, 1}
584	{5, 5, 0}	620	{6, 5, 2}	656	{5, 3, 7}	692	{11, 5, 10}
585	{2, 1, 1}	621	{14, 38, 14}	657	{5, 1, 8}	693	{16, 25, 38}
586	{9, 18, 19}	622	{10, 13, 17}	658	{9, 5, 15}	694	{14, 5, 22}
587	{15, 20, 35}	623	{24, 12, 8}	659	{11, 11, 24}	695	{12, 20, 35}
588	{4, 3, 2}	624	{10, 12, 16}	660	{3, 1, 2}	696	{12, 34, 2}
589	{12, 15, 27}	625	{12, 11, 20}	661	{4, 6, 1}	697	{11, 28, 8}
590	{17, 57, 10}	626	{8, 8, 10}	662	{6, 10, 7}	698	{21, 19, 5}
591	{26, 91, 24}	627	{8, 11, 14}	663	{15, 3, 32}	699	{20, 28, 44}
592	{6, 3, 7}	628	{13, 30, 11}	664	{9, 7, 3}	700	{10, 26, 10}
593	{6, 1, 8}	629	{11, 3, 12}	665	{11, 20, 14}	701	{17, 28, 38}
594	{3, 1, 0}	630	{10, 27, 16}	666	{7, 5, 3}	702	{23, 77, 36}
595	{7, 5, 1}	631	{18, 6, 8}	667	{14, 13, 42}	703	{20, 56, 1}
596	{4, 1, 2}	632	{11, 12, 22}	668	{11, 17, 29}	704	{6, 11, 0}
597	{9, 20, 1}	633	{24, 47, 20}	669	{11, 13, 6}	705	{6, 5, 8}
598	{7, 14, 7}	634	{14, 20, 10}	670	{16, 12, 49}	706	{5, 8, 0}
599	{15, 13, 1}	635	{20, 61, 54}	671	{18, 24, 27}	707	{7, 13, 8}
600	{5, 8, 2}	636	{18, 24, 40}	672	{8, 3, 10}	708	{11, 24, 7}
601	{11, 18, 20}	637	{15, 41, 42}	673	{10, 26, 6}	709	{10, 18, 20}
602	{8, 14, 0}	638	{22, 41, 21}	674	{5, 10, 2}	710	{14, 35, 0}
603	{10, 2, 12}	639	{28, 55, 84}	675	{11, 28, 4}	711	{12, 29, 6}
604	{15, 3, 26}	640	{7, 15, 15}	676	{5, 1, 10}	712	{6, 8, 10}
605	{21, 63, 44}	641	{5, 9, 4}	677	{4, 1, 6}	713	{12, 22, 14}
606	{13, 26, 16}	642	{7, 5, 7}	678	{17, 9, 26}	714	{7, 12, 2}
607	{26, 95, 30}	643	{11, 25, 12}	679	{11, 17, 1}	715	{13, 37, 1}
608	{9, 11, 11}	644	{9, 18, 7}	680	{10, 3, 26}	716	{14, 34, 19}
609	{6, 6, 4}	645	{10, 24, 1}	681	{9, 3, 20}	717	{20, 11, 32}
610	{5, 6, 9}	646	{6, 3, 11}	682	{1, 1, 0}	718	{13, 18, 29}
611	{12, 18, 27}	647	{6, 4, 4}	683	{23, 34, 8}	719	{14, 20, 27}

Table 2. (*continues*).

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
720	{7, 15, 9}	756	{17, 28, 41}	792	{3, 3, 2}	828	{11, 12, 3}
721	{8, 7, 14}	757	{17, 31, 54}	793	{12, 19, 4}	829	{14, 20, 30}
722	{8, 5, 7}	758	{23, 73, 64}	794	{12, 8, 0}	830	{20, 41, 16}
723	{9, 7, 1}	759	{20, 6, 18}	795	{13, 18, 14}	831	{25, 47, 80}
724	{11, 26, 3}	760	{14, 6, 31}	796	{7, 6, 11}	832	{5, 9, 2}
725	{12, 30, 32}	761	{15, 6, 35}	797	{13, 3, 18}	833	{7, 5, 12}
726	{17, 46, 11}	762	{13, 35, 7}	798	{12, 29, 22}	834	{10, 5, 19}
727	{17, 19, 48}	763	{20, 65, 48}	799	{13, 12, 35}	835	{8, 14, 6}
728	{8, 17, 7}	764	{17, 56, 10}	800	{6, 8, 7}	836	{8, 7, 0}
729	{14, 5, 12}	765	{31, 80, 32}	801	{6, 7, 8}	837	{14, 33, 12}
730	{9, 21, 3}	766	{35, 134, 21}	802	{7, 10, 13}	838	{14, 9, 9}
731	{9, 2, 6}	767	{23, 68, 4}	803	{8, 9, 18}	839	{10, 21, 20}
732	{7, 2, 0}	768	{7, 2, 5}	804	{5, 5, 5}	840	{5, 3, 10}
733	{11, 4, 27}	769	{6, 2, 1}	805	{11, 20, 30}	841	{15, 27, 27}
734	{19, 8, 49}	770	{5, 2, 10}	806	{4, 4, 0}	842	{6, 3, 3}
735	{25, 15, 27}	771	{9, 11, 1}	807	{19, 37, 20}	843	{18, 50, 4}
736	{5, 2, 5}	772	{6, 5, 11}	808	{10, 11, 23}	844	{9, 7, 10}
737	{12, 34, 8}	773	{11, 6, 20}	809	{7, 12, 12}	845	{13, 4, 6}
738	{10, 8, 16}	774	{8, 4, 10}	810	{11, 14, 16}	846	{11, 13, 19}
739	{13, 38, 4}	775	{8, 15, 12}	811	{17, 50, 57}	847	{25, 8, 93}
740	{10, 2, 23}	776	{5, 4, 5}	812	{11, 18, 7}	848	{10, 4, 11}
741	{11, 28, 14}	777	{5, 8, 6}	813	{9, 16, 18}	849	{11, 22, 6}
742	{13, 11, 13}	778	{6, 10, 2}	814	{20, 44, 7}	850	{7, 3, 11}
743	{19, 24, 20}	779	{8, 5, 18}	815	{13, 26, 1}	851	{17, 9, 35}
744	{11, 6, 9}	780	{12, 11, 10}	816	{6, 6, 10}	852	{9, 3, 0}
745	{21, 10, 27}	781	{7, 4, 6}	817	{7, 8, 8}	853	{23, 25, 1}
746	{20, 43, 40}	782	{4, 2, 2}	818	{6, 7, 10}	854	{16, 25, 7}
747	{28, 98, 51}	783	{9, 6, 6}	819	{19, 16, 38}	855	{17, 28, 4}
748	{10, 16, 5}	784	{5, 7, 0}	820	{14, 19, 9}	856	{14, 8, 3}
749	{11, 27, 1}	785	{9, 13, 6}	821	{14, 9, 6}	857	{17, 34, 24}
750	{18, 55, 50}	786	{5, 6, 2}	822	{13, 23, 22}	858	{17, 40, 2}
751	{32, 110, 93}	787	{8, 10, 4}	823	{14, 23, 32}	859	{18, 53, 38}
752	{10, 26, 16}	788	{6, 4, 11}	824	{10, 12, 28}	860	{15, 9, 17}
753	{12, 8, 32}	789	{11, 3, 8}	825	{8, 8, 18}	861	{28, 9, 77}
754	{20, 34, 10}	790	{12, 11, 13}	826	{13, 30, 0}	862	{24, 68, 25}
755	{14, 23, 38}	791	{10, 8, 27}	827	{18, 45, 54}	863	{22, 15, 27}

Table 2. (*continues*).

https://doi.org/10.25088/ComplexSystems.22.4.405

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
864	{6, 5, 0}	900	{5, 4, 9}	936	{13, 28, 36}	972	{14, 31, 28}
865	{12, 29, 4}	901	{6, 9, 1}	937	{15, 34, 24}	973	{14, 23, 35}
866	{11, 24, 0}	902	{7, 4, 10}	938	{10, 3, 3}	974	{11, 19, 11}
867	{13, 23, 12}	903	{4, 5, 1}	939	{17, 31, 20}	975	{32, 60, 68}
868	{6, 8, 5}	904	{5, 7, 7}	940	{15, 38, 22}	976	{13, 6, 11}
869	{14, 30, 30}	905	{12, 27, 20}	941	{17, 40, 42}	977	{14, 28, 18}
870	{16, 24, 47}	906	{6, 4, 2}	942	{21, 23, 39}	978	{19, 53, 49}
871	{13, 18, 6}	907	{10, 8, 6}	943	{32, 9, 110}	979	{13, 15, 30}
872	{7, 15, 7}	908	{7, 11, 9}	944	{6, 11, 9}	980	{15, 34, 11}
873	{10, 27, 1}	909	{12, 8, 8}	945	{12, 20, 6}	981	{10, 3, 6}
874	{12, 17, 13}	910	{8, 6, 2}	946	{10, 16, 2}	982	{22, 64, 19}
875	{18, 8, 27}	911	{13, 12, 6}	947	{14, 20, 24}	983	{32, 122, 60}
876	{8, 17, 5}	912	{8, 8, 5}	948	{7, 15, 5}	984	{12, 29, 25}
877	{9, 21, 1}	913	{11, 17, 6}	949	{12, 17, 30}	985	{24, 77, 27}
878	{8, 2, 2}	914	{17, 2, 49}	950	{8, 17, 3}	986	{19, 52, 13}
879	{19, 8, 57}	915	{19, 30, 18}	951	{25, 15, 57}	987	{19, 59, 62}
880	{6, 2, 9}	916	{14, 4, 19}	952	{14, 37, 19}	988	{16, 47, 13}
881	{11, 2, 27}	917	{11, 28, 20}	953	{18, 55, 1}	989	{32, 21, 35}
882	{10, 2, 25}	918	{11, 18, 25}	954	{20, 34, 26}	990	{17, 53, 7}
883	{18, 16, 32}	919	{22, 56, 27}	955	{18, 6, 51}	991	{17, 6, 6}
884	{11, 4, 19}	920	{8, 11, 0}	956	{17, 53, 13}	992	{10, 23, 2}
885	{20, 43, 12}	921	{10, 17, 27}	957	{20, 65, 1}	993	{16, 41, 8}
886	{10, 13, 7}	922	{14, 18, 9}	958	{16, 6, 0}	994	{16, 6, 29}
887	{18, 55, 44}	923	{13, 23, 4}	959	{32, 110, 4}	995	{13, 29, 20}
888	{12, 8, 3}	924	{11, 19, 23}	960	{9, 6, 16}	996	{15, 6, 23}
889	{20, 10, 35}	925	{19, 43, 35}	961	{10, 23, 8}	997	{18, 58, 14}
890	{20, 65, 7}	926	{11, 12, 15}	962	{11, 8, 10}	998	{20, 30, 0}
891	{27, 14, 51}	927	{25, 47, 30}	963	{9, 17, 14}	999	{25, 50, 65}
892	{15, 6, 41}	928	{5, 9, 0}	964	{12, 23, 5}	1000	{13, 6, 17}
893	{20, 6, 65}	929	{14, 33, 8}	965	{12, 3, 14}	1001	{26, 6, 77}
894	{25, 15, 94}	930	{13, 22, 37}	966	{12, 8, 16}	1002	{20, 15, 64}
895	{23, 68, 72}	931	{10, 21, 12}	967	{13, 29, 32}	1003	{22, 64, 4}
896	{6, 2, 3}	932	{15, 37, 0}	968	{15, 6, 17}	1004	{15, 41, 13}
897	{5, 2, 1}	933	{15, 44, 12}	969	{26, 10, 35}	1005	{25, 82, 65}
898	{7, 6, 15}	934	{13, 15, 15}	970	{16, 41, 10}	1006	{16, 47, 7}
899	{8, 4, 14}	935	{22, 23, 57}	971	{25, 89, 68}	1007	{17, 53, 1}

Table 2. (*continues*).

https://doi.org/10.25088/ComplexSystems.22.4.405

RN	Tuples	RN	Tuples	RN	Tuples	RN	Tuples
1008	{17, 56, 16}	1012	{17, 3, 36}	1016	{23, 71, 7}	1020	{23, 68, 34}
1009	{21, 35, 1}	1013	{20, 15, 8}	1017	{28, 52, 44}	1021	{23, 15, 35}
1010	{22, 38, 36}	1014	{23, 15, 5}	1018	{26, 15, 53}	1022	{30, 98, 19}
1011	{25, 50, 18}	1015	{32, 21, 77}	1019	{23, 15, 20}	1023	{1, 0, 1}

Table 2. Iota-delta tuples $\{m, \alpha_1, \alpha_2\}$ for totalistic CAs with the Neumann neighborhood: first part.

4. The Iota-Delta Function and Outer Totalistic Two-Dimensional Cellular Automata

On the other hand, by definition, outer totalistic CAs uniquely determine the value of a given cell in the next time step by the cell's value and the sum of the cells in its neighborhood template, excluding from the sum the cell itself [1]. This way, as in the case of totalistic CAs, the number of cells in the neighborhood template is not important since the value of a given cell in the next time step is dependent on only two variables, namely: the value of the cell itself and the sum of its neighbors. This way, the iota-delta evolution rule is of the form:

$$C_{k,j}^{i+1} = \iota\delta_2^m \left[\alpha_1 \overline{\sigma} + \alpha_2 C_{k,j}^i + \alpha_3\right], \tag{5}$$

in which $\overline{\sigma}$ is the value of the sum of the neighbors of the cell of interest, excluding the value of the latter ($C_{k,j}^i$). An equation of the form of equation (5) is applicable to every outer totalistic two-dimensional CA with any type of neighborhood template. In the case of a Neumann neighborhood, a total of 2^{10} outer totalistic rules are available. On the other hand, for a Moore neighborhood, an astronomical 2^{18} different rules exist. Since the complete lists of iota-delta representations of totalistic two-dimensional CAs with both Neumann and Moore neighborhoods have been presented, the lists concerning outer totalistic rules are omitted. Section 5 shows that by knowing the iota-delta representations of a totalistic CA, the rules for outer totalistic representations easily follow.

Even though the complete list of outer totalistic rules is suppressed, one of the most famous two-dimensional CAs of this class is discussed in Section 4.1: the Game of Life [3].

4.1 A Special Two-Dimensional Outer Totalistic Cellular Automaton with the Moore Neighborhood: Game of Life

By means of equation (5), the Game of Life CA (rule 224 following the numbering system defined in [1]) has its evolution rule described

https://doi.org/10.25088/ComplexSystems.22.4.405

in terms of the iota-delta function as:

$$C_{k,j}^{i+1} = \iota\delta_2^{12}\left[14\,\overline{\sigma} + 6\,C_{k,j}^{i} + 9\right]. \tag{6}$$

Despite the apparent simplicity of equation (6), applying it to a two-dimensional lattice causes the well-known Game of Life patterns to emerge. Section 5 introduces the concept that the iota-delta function is a basis for two-dimensional CA transition rule space.

5. The Iota-Delta Function as a Basis for Two-Dimensional Cellular Automata

In order to better understand this section, consider the possible inputs and outputs of the evolution of a two-dimensional totalistic CA with a Neumann neighborhood, as given in Table 3.

Inputs (σ)	0	1	2	3	4	5
Outputs	β_1	β_2	β_3	β_4	β_5	β_6

Table 3. Inputs and outputs of the evolution rules of totalistic CAs with a Neumann neighborhood.

It is clear from Table 3 that the number of a given rule is simply given as:

$$\mathrm{RN} = \sum_{w=1}^{6} 2^{w-1}\,\beta_w. \tag{7}$$

As a special case, consider the inputs and outputs of the evolution of the rules 2^q, $q = 0, 1, \ldots, 5$, as in Table 4.

Inputs (σ)	0	1	2	3	4	5	RN
Outputs	1	0	0	0	0	0	1
	0	1	0	0	0	0	2
	0	0	1	0	0	0	4
	0	0	0	1	0	0	8
	0	0	0	0	1	0	16
	0	0	0	0	0	1	32

Table 4. Inputs and outputs of the 2^q, $q = 0, 1, \ldots, 5$, evolution rules of totalistic CAs with a Neumann neighborhood.

https://doi.org/10.25088/ComplexSystems.22.4.405

From Table 4 it is possible to notice that each of the rules described has a unitary output for only one of the given input states. This way, the iota-delta representation of the evolution of each rule in Table 4 has the property of a unitary Lagrange polynomial, which is 1 in a given point and 0 in others.

Unitary Lagrange polynomials are the basis for interpolating polynomials; that is, by means of linearly combining the unitary Lagrange polynomials, every interpolating polynomial for a given set of data is obtained.

By drawing a parallel between interpolating polynomials and CA evolution rules, then every evolution rule can be expressed as a combination of the iota-delta bases. The iota-delta bases are, on the other hand, the iota-delta representation of the evolution of rules 2^q, $q = 0, 1, \ldots, T$, in which T is the total number of inputs. Mathematically, let $\underline{\beta}$ be a vector whose components are the outputs of the evolution rule. Also, let $\underline{\iota\delta}$ be the vector whose components are the basis evolution rules. Thus, every evolution rule (ER in the equations) is the projection of $\underline{\iota\delta}$ over $\underline{\beta}$ (dot product). This result is easily expressed as:

$$\text{ER}\left[\underline{\beta}, T\right] = \underline{\iota\delta}.\underline{\beta} = \sum_{w=1}^{T} \beta_w \, \iota\delta_2^{m_w} \left[\sum_{s=1}^{Q} \alpha_{w,s} \, v_s + \alpha_{w,Q+1}\right], \tag{8}$$

in which Q is the number of variables on which the evolution rule depends. In the case of totalistic CAs, $Q = 1$. On the other hand, for outer totalistic CAs $Q = 2$, and for elementary CAs $Q = 3$. Also, v_j is the j^{th} variable on which the evolution rule depends, $j = 1, \ldots, Q$. Finally, $\alpha_{q,j}$ is the coefficient of the j^{th} variable on which the iota-delta evolution rule of rule 2^q depends, $q = 0, 1, \ldots, T$.

In order to illustrate the applicability of the methodology described, two extended neighborhood templates are considered in Sections 5.1 and 5.2.

5.1 Iota-Delta Bases for Two-Dimensional Cellular Automata with Extended Neumann and Moore Neighborhoods

Consider Figure 1, in which both extended Neumann and Moore neighborhoods are presented.

The σ values range from 0 to 13 and from 0 to 25 for extended Neumann and Moore neighborhoods, respectively. This means we have to find 14 and 26 iota-delta bases for each, respectively. Table 5 shows the coefficients for obtaining the bases for both extended Neumann and Moore neighborhoods.

https://doi.org/10.25088/ComplexSystems.22.4.405

RN	Tuples EN	Tuples EM	RN	Tuples EN	Tuples EM
1	{13, 3, 4}	{22, 3, 4}	8192	{13, 38, 2}	{30, 103, 23}
2	{9, 2, 22}	{20, 68, 11}	16384	x	{30, 103, 33}
4	{10, 27, 2}	{30, 107, 47}	32768	x	{27, 93, 53}
8	{11, 29, 2}	{28, 101, 53}	65536	x	{23, 73, 0}
16	{13, 24, 39}	{26, 95, 59}	131072	x	{23, 73, 10}
32	{11, 6, 5}	{32, 128, 5}	262144	x	{27, 93, 83}
64	{12, 8, 7}	{30, 10, 59}	524288	x	{30, 103, 83}
128	{12, 29, 0}	{27, 10, 39}	1048576	x	{32, 3, 61}
256	{11, 25, 21}	{23, 10, 9}	2097152	x	{26, 6, 10}
512	{13, 17, 23}	{23, 10, 82}	4194304	x	{28, 6, 10}
1024	{11, 2, 7}	{27, 10, 9}	8388608	x	{30, 6, 10}
2048	{10, 2, 5}	{30, 10, 9}	16777216	x	{20, 3, 7}
4096	{9, 21, 2}	{30, 10, 112}	33554432	x	{22, 76, 0}

Table 5. Tuples of coefficients $\{m, \alpha_1, \alpha_2\}$ for obtaining the bases for both extended Neumann (EN) and Moore (EM) neighborhoods.

The values in Table 5 can be inserted in equation (8) in order to provide every totalistic two-dimensional CA with both extended Neumann and Moore neighborhoods. Also, let T_1 be the number of inputs for a given CA. Now, let T_2 be the number of inputs for another given CA with the same number of variables ($Q_1 = Q_2$). Thus, if $T_2 \geq T_1$, then the first T_1 bases of the second CA can also be used as bases for the first one. This follows from the Lagrange polynomial property of the iota-delta bases.

5.2 Iota-Delta Bases for Two-Dimensional Outer Totalistic Cellular Automata from Bases for Totalistic Ones

We now wonder whether or not bases for totalistic CAs can be used to obtain outer totalistic CAs. In fact, this can be easily accomplished by considering the evolution rule of the outer totalistic CA as the sum of two evolution rules for totalistic CAs. Each of the totalistic rules corresponds to a hypothetical rule in which the cell value $C_{k,j}^i$ is kept constant. Mathematically, the outer totalistic rule is given as:

$$\text{ER}_{\text{outer}}\left[\underline{\beta}, T\right] = \left(1 - C_{k,j}^i\right) \text{ER}\left[\underline{\beta}_0, T-1\right] + C_{k,j}^i \,\text{ER}\left[\underline{\beta}_1, T-1\right], \tag{9}$$

in which β_0 is the output values vector when $C_{k,j}^i = 0$, and β_1 is the output values vector when $C_{k,j}^i = 1$. Since the central value in the tem-

https://doi.org/10.25088/ComplexSystems.22.4.405

plate has been taken out of the summation, each of the remaining evolution rules turns out to have only $T - 1$ input values. Also, based on the fact that bases for CAs with more inputs than a given one can be used as bases for the latter, the totalistic bases suit the need.

As an example, consider the Game of Life evolution rule. It has been shown that such an evolution rule is expressible as in equation (6). On the other hand, consider Table 6, in which the outer totalistic rule is separated into two totalistic rules.

	Inputs ($\overline{\sigma}$)									
Outputs when $C^i_{k,j}$ equals	0	1	2	3	4	5	6	7	8	RN
0	0	0	0	1	0	0	0	0	0	8
1	0	0	1	1	0	0	0	0	0	12

Table 6. Game of Life decomposition.

By means of Table 6, it is easily seen that the Game of Life is a combination of rules 8 and 12. Rule 8 is a basis itself. Rule 12, on the other hand, is the combination of two bases, namely: the basis concerning rule 4 and that concerning rule 8. Thus, by means of equation (9) and Table 2, the evolution of the Game of Life CA is given as:

$$\begin{aligned} C^{i+1}_{k,j} &= \left(1 - C^i_{k,j}\right) \mathrm{ER}_8 + C^i_{k,j}\, \mathrm{ER}_{12} = \\ &\left(1 - C^i_{k,j}\right) \mathrm{ER}_8 + C^i_{k,j} (\mathrm{ER}_4 + \mathrm{ER}_8) = \\ &\mathrm{ER}_8 + C^i_{k,j}\, \mathrm{ER}_4\, \iota\delta^8_2\, [6\,\overline{\sigma} + 5] + C^i_{k,j}\, \iota\delta^9_2\, [6\,\overline{\sigma} + 15]. \end{aligned} \tag{10}$$

The given procedure can be extended to every other two-dimensional CA by finding the bases of such a space. Thus, any two-dimensional CA can be evolved by means of an iota-delta rule.

6. Conclusions

In the last half century, cellular automata (CAs) have been deeply studied by the scientific community. In general, researchers have been worried about the patterns produced by a given CA rule. The latter were taken into account only to implement the former, but not as a true information source about the phenomena simulated. In continuous dynamical systems, partial differential equations are the evolution rules and also carry important information about the phenomena under study.

In the present paper, the iota-delta function has been used to express the evolution rules of two-dimensional totalistic and outer totalistic CAs with a set of neighborhood templates. Also, the concept of iota-delta bases for the evolution rules space has been introduced. This latter concept enables representing every two-dimensional CA in terms of suitable iota-delta bases.

Acknowledgments

The authors would like to thank the University of Brasilia (UnB), the Brazilian National Research Council (CNPq), and the Coordination for the Improvement of Higher Level Personnel (CAPES) for funding this research.

References

[1] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[2] D. Gordon, "On the Computational Power of Totalistic Cellular Automata," *Mathematical Systems Theory*, **20**, 1987 pp. 43–52. doi:10.1007/BF01692058.

[3] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning Ways for Your Mathematical Plays*, Vol. 2, London, New York: Academic Press, 1982.

[4] M. Mitchell, J. P. Crutchfield, and R. Das, "Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work," in *Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96)*, Moscow, 1996, Moscow: Russian Academy of Sciences.

[5] L. C. de S. M. Ozelim, A. L. B. Cavalcante, and L. P. de F. Borges, "On the Iota-Delta Function: Universality in Cellular Automata's Representation," *Complex Systems*, **21**(4), 2013 pp. 283–296. http://www.complex-systems.com/pdf/21-4-3.pdf.

https://doi.org/10.25088/ComplexSystems.22.4.405